

**REMARKS**

Claims 1, 3, 6-11, 14-20, and 24-31 are pending in the present application as preliminarily amended. Claims 2, 4, 5, 21-23 and 32-50 have been canceled. Independent claim 1 has been amended to include the subject matter of now-canceled claims 2, 4, and 5, and independent claim 20 has likewise been amended to include the subject matter of now-canceled claims 21-23. Applicants respectfully submit that no new matter has been added to the application by the amendment.

Claims 1, 3, 6-11, 14-15, 18-20, and 24-29 stand rejected under 35 U.S.C. § 103(a) as being obvious over U.S. Patent No. 6,158,045 (You). Applicants respectfully traverse the You rejection insofar as it may be applied to the claims as amended.

Independent claim 1 recites a debugger for debugging any of a plurality of debuggees. Each debuggee has a debugging type attribute selected from a plurality of debugging type attributes and representative of a type of debugging to be performed with respect to the debuggee, and each debuggee also has a processor attribute selected from a plurality of processor attributes and representative of a type of processor associated with the debuggee. The debugger is instantiated on a computer and has a single debugger engine for performing debugging functions with respect to any of the plurality of debuggees.

The engine includes a plurality of debugging type blocks, where each debugging type block supports at least one of the plurality of debugging type attributes, and a plurality of processor blocks, where each processor block supports at least one of the plurality of processor attributes. A particular debugging type block and a particular processor block are selected for debugging a particular debuggee based on the debugging type attribute and processor attribute of the particular debuggee.

Significantly, the plurality of debugging type blocks are organized into a debugging type abstraction available to provide debugging type services that vary in implementation for each debugging type. The debugging type abstraction comprises programming code, and at least a portion of the programming code for the debugging type abstraction is common as between at least some debugging type blocks and is shared by such debugging type blocks. In particular, the programming code for the debugging type abstraction is organized into a tree form with generic code at a base node and more specific levels of code branching out at nodes therefrom. Thus, each debugging type block includes at least one node from the tree.

Independent claim 20 recites subject matter similar to that of claim 1, albeit in the form of a computer with the debugger instantiated thereon.

Although the You reference does disclose a debugger portable to multiple operating systems and hardware platforms, Applicants respectfully submit that the You reference does not disclose or suggest that a plurality of debugging type blocks, each representing a debugging type, be organized into a debugging type abstraction available to provide debugging type services that vary in implementation for each debugging type, where the debugging type abstraction comprises programming code that is organized into a tree form with generic code at a base node and more specific levels of code branching out at nodes therefrom, all as is now required by claims 1 and 20.

Although the Examiner points to Fig. 9 of the You reference as showing such a tree form of an abstraction, Applicants respectfully point out that such Fig. 9 in fact shows an addressing abstraction utilized to facilitate the use of target memory addresses in a portable fashion (Abstract), and that such an addressing abstraction is not at all the debugging type abstraction as recited in claims 1 and 20. In particular, the target memory address abstraction contains classes and sub-classes necessary to locate various addresses depending on operating

system and/or platform. In contrast, the debugging type abstraction of claims 1 and 20 contains code necessary to define multiple types of debugging type blocks, where such code is organized in a tree form from more generic code to more specific code. Moreover, Applicants respectfully submit that the target memory address abstraction shown in Fig. 9 of the You reference does not even suggest the debugging type abstraction, as is recited in claims 1 and 20, especially inasmuch as the You reference contains no suggestion or hint that by providing such a debugging type abstraction, multiple types of debuggees can be handled by a single debugger.

Thus, Applicants respectfully submit that the You reference cannot be applied to make obvious claims 1 or 20 or any claims depending therefrom. As a result, Applicants respectfully request reconsideration and withdrawal of the You rejection.

Claims 16-17 and 30-31 stand rejected under § 103(a) as being obvious over the You reference in view of U.S. Patent No. 5,533,192 (Hawley et al.). Applicants respectfully traverse the You-Hawley rejection insofar as it may be applied to the claims as amended.

Applicants respectfully point out that since independent claims 1 and 20 are unanticipated and have been shown to be non-obvious, then so too must all claims depending therefrom including claims 16-17 and 30-31 be unanticipated and non-obvious, at least by their dependency. As a result, Applicants respectfully request reconsideration and withdrawal of the You-Hawley rejection.

**DOCKET NO.:** MSFT-0218/146820.01

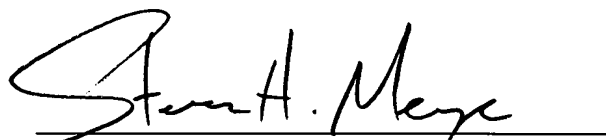
**PATENT**

**Application No.:** 09/681,064

**Preliminary Amendment -**

In view of the foregoing, Applicants respectfully submit that the present application including claims 1, 3, 6-11, 14-20, and 24-31 is in condition for allowance, and such action is respectfully requested.

Respectfully submitted,

A handwritten signature in black ink, appearing to read "Steven H. Meyer", written over a horizontal line.

Steven H. Meyer  
Registration No. 37,189

Date: March 14, 2005

Woodcock Washburn LLP  
One Liberty Place - 46th Floor  
Philadelphia PA 19103  
Telephone: (215) 568-3100  
Facsimile: (215) 568-3439